**WHAT IS CLAIMED IS:**

1. A method of preventing upload overloads of data from a plurality of clients at different locations within a network to a common destination server in the network, the steps comprising:

generating a unique identifier corresponding to and dependent on data that each client intends to send to the common destination server, the unique identifier being smaller in size than the data of the client;

separately transmitting the unique identifiers from each client to at least one authenticator trusted by the common destination server;

separately time-stamping the unique identifiers as received by the authenticator;

separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client and the corresponding time-stamp;

each client then sending its data towards the common destination server; and

the common destination server using the unique identifier for the data provided by each client to confirm that the data provided by each client existed as of the corresponding time-stamp and to insure that the data has been unaltered after the corresponding time-stamp.

2. The method of Claim 1 wherein the generating of the unique identifier is accomplished by using a one-way hash function on the data of each client.

3. The method of Claim 1 wherein the sending of data towards the common destination server by each client is accomplished by sending the data to one of a plurality of upload proxy servers.

4. The method of Claim 3 further comprising the step of: having each upload proxy server send a message to the common destination server indicating that it is holding data for it.

5. The method of Claim 4 further comprising the step of having the common destination server upload the data held for it at any one or more of the upload proxy servers.

6. The method of Claim 5 wherein, prior to the step of generating the unique identifiers, a sponsor anticipating that a plurality of clients will send large amounts of different data intended for the common destination server in a relatively short time interval, performs the steps of:

establishing the authenticator for the anticipated large amounts of data; and

supplying to the authenticator criteria for receiving data from the plurality of clients.

7. The method of Claim 6 further comprising the step of having the authenticator create an event identifier (EID) corresponding to the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time interval and wherein the authenticator publishes the EID before the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time interval.

8. The method of Claim 1 wherein, prior to the step of generating the unique identifiers, a sponsor anticipating that a plurality of clients will send large amounts of different data intended for the common destination server in a relatively short time interval, performs the steps of:

establishing the authenticator for the anticipated large amounts of data; and

supplying to the authenticator criteria for receiving data from the plurality of clients.

9. The method of Claim 8 further comprising the step of having the authenticator create an event identifier (EID) corresponding to the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time interval and wherein the authenticator publishes the EID before the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time.

10. The method of Claim 8 wherein the criteria for receiving data includes an encryption level to be used when each client is sending data towards the common destination server.

11. The method of Claim 3 further comprising the step of: having each upload proxy server send a message acknowledging receipt of data sent to it by a client.

12. The method of Claim 3 further comprising the step of having the authenticator send a message to a client containing a designation of at least one upload proxy server for use by the client.

13. A method of preventing upload overloads of data from a plurality of clients at

different locations within a network to a common destination server in the network, the steps comprising:

generating a unique identifier corresponding to and dependent on data that each client intends to send to the common destination server, the unique identifier being smaller in size than the data of the client;

separately transmitting the unique identifiers from each client to at least one authenticator trusted by the common destination server;

separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client;

each client then forwarding its data to the common destination server via proxy upload servers remote from the common destination server; and

the common destination server using the unique identifier for the data provided by each client to confirm that the data provided by each client has been unaltered after the generation of the unique identifier.

14. The method of Claim 13 wherein the unique identifiers are one-way hashes of the data that they correspond to.

15. The method of Claim 13 further comprising the step of having each upload proxy server send a message to the common destination server indicating that it is holding data for it after that upload proxy server has received data from a client.

16. The method of Claim 15 further comprising the step of uploading data from at least one of the upload proxy servers to the common destination server

responsive to messages sent indicating that the particular upload proxy server is holding data for the common destination server.

17. The method of Claim 16 further comprising the step of separately time-stamping the unique identifiers as received by the authenticator; and wherein the step of separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client includes the corresponding time-stamp within the message; and wherein

the common destination server uses the unique identifier for the data provided by each client to confirm that the data provided by each client existed as of the corresponding time-stamp and to insure that the data has been unaltered after the corresponding time-stamp.

18. The method of Claim 16 wherein, prior to the step of generating the unique identifiers, a sponsor anticipating that a plurality of clients will send large amounts of different data intended for the common destination server in a relatively short time interval, performs the steps of:

establishing the authenticator for the anticipated large amounts of data; and

supplying to the authenticator criteria for receiving data from the plurality of clients.

19. The method of Claim 18 further comprising the step of having the authenticator create an event identifier (EID) corresponding to the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time interval and wherein the authenticator

publishes the EID before the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time.

20. The method of Claim 19 wherein the criteria for receiving data includes an encryption level to be used when each client is sending data towards the common destination server.

21. The method of Claim 13 further comprising the step of separately time-stamping the unique identifiers as received by the authenticator; and wherein the step of separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client includes the corresponding time-stamp within the message; and wherein

the common destination server uses the unique identifier for the data provided by each client to confirm that the data provided by each client existed as of the corresponding time-stamp and to insure that the data has been unaltered after the corresponding time-stamp.

22. The method of Claim 13 wherein, prior to the step of generating the unique identifiers, a sponsor anticipating that a plurality of clients will send large amounts of different data intended for the common destination server in a relatively short time interval, performs the steps of:

establishing the authenticator for the anticipated large amounts of data; and

supplying to the authenticator criteria for receiving data from the plurality of clients.

23. A method of preventing upload overloads of data from a plurality of clients at

different locations within a network to a common destination server in the network, the steps comprising:

providing a common destination server in a network, the common destination server set up to receive data from a plurality of clients:

providing a plurality of upload proxy servers remote from the common destination server;

each client sending data, which is intended for the common destination server, to at least a corresponding one of the upload proxy servers;

sending a message, which is smaller in size than the data of a client, to the common destination server to indicate that the common destination server needs to check the corresponding one of the upload proxy servers; and

having the common destination server upload the data of a client at some time after the message such that a plurality of clients trying to send data to the common destination server at essentially the same time is less likely to overload the common destination server and its connection to the network.

24. The method of Claim 23 wherein, prior to the step of the clients sending data, a sponsor anticipating that a plurality of clients will send large amounts of different data intended for the common destination server in a relatively short time interval, performs the steps of:

establishing an authenticator for the anticipated large amounts of data; and

supplying to the authenticator criteria for receiving data from the plurality of clients.

25. The method of Claim 24 further comprising the step of having the authenticator create an event identifier (EID) corresponding to the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time interval and wherein the authenticator publishes the EID before the anticipated plurality of clients sending large amounts of different data intended for the common destination server in a relatively short time.

26. The method of Claim 25 wherein the criteria for receiving data includes an encryption level to be used when each client is sending data towards the common destination server.

27. The method of Claim 23 further comprising the steps of, prior to each client sending data to at least one of the upload proxy servers:

generating a unique identifier corresponding to and dependent on data that each client intends to send to the common destination server, the unique identifier being smaller in size than the data of the client;

separately transmitting the unique identifiers from each client to at least one authenticator trusted by the common destination server; and

separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client; and

having the common destination server use the unique identifier for the data provided by each client to confirm that the data provided by each client has been unaltered after the generation of the unique identifier.

28. The method of Claim 23 further comprising the step of separately time-stamping the unique identifiers as received by the authenticator; and wherein the step of separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client includes the corresponding time-stamp; and having the common destination server use the unique identifier for the data provided by each client to confirm that the data provided by each client existed as of the corresponding time-stamp and to insure that the data has been unaltered after the corresponding time-stamp.

29. A system for preventing upload overloads of data from a plurality of clients at different locations within a network to a common destination server in the network, comprising:

a common destination server in a network, the common destination server set up to receive data from a plurality of clients:

an id generator operable to generate a unique identifier corresponding to and dependent on data that each client intends to send to the common destination server, the unique identifier being smaller in size than the data of the client;

each client having a sender for separately transmitting the unique identifier from that client;

at least one authenticator trusted by the common destination server, the authenticator having a time-stamper for separately time-stamping the unique identifiers as received by the authenticator, the authenticator having a sender for separately sending back to each client a message, digitally signed by the authenticator, with the unique identifier sent by that client and the corresponding time-stamp; and

wherein the common destination server includes a checker that uses the unique identifier for the data provided by each client to confirm that the data provided by each client existed as of the corresponding time-stamp and to insure that the data has been unaltered after the corresponding time-stamp.

30. The system of Claim 29 wherein each client is operable to send the data towards the common destination server after receiving the message from the authenticator.

31. The system of Claim 30 further comprising a plurality of upload proxy servers operable to receive the data provided by each client and wherein the common destination server is operable to upload data held for it by the upload proxy servers.

32. The system of Claim 31 wherein the id generator takes a one-way hash of the data that the client intends to send to the common destination server.